#9

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

U.S. Patent Application No.: 09/699,523

Filing Date: October 30, 2000

For: Random Number Generator And Generation Method

Applicants: Scott A. Wilber

Group Art Unit: 2124

Examiner: Mai, Tan V.

Docket No: 13323.102
(Formerly 2022/002D1)

DECLARATION OF SCOTT A. WILBER

I, Scott A. Wilber, hereby declare:

1. I am President and founder of The Quantum World Corporation, having a mailing address at P.O. Box 370, Rosewell, NM, 88202. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true.

2. I have worked for twenty-seven years as an electronics expert, an entrepreneur and inventor, founding five companies. My most significant invention prior to the present application was the pulse oximeter, which is described in United States Patent No. 4,407,290, which was purchased by British Oxygen Corporation and now is the basis of a billion dollar industry. I have published a half-dozen papers in chemistry and solid-state physics.

3. By virtue of the fact that I have studied almost every reference available in the random number generator art, have designed one of the most effective random number generators ever built, and have been completely immersed in the field for ten years, I am highly skilled in the true random number generator art.

4. I am the sole inventor of the invention described and claimed in the above-identified United States Patent Application Serial No. 09/699,523 (hereinafter, "the application").

5. I have read the claims at issue, namely claims 57 – 66 of the application (hereinafter, "the claims"), a copy of which is attached as Exhibit A. I have also read the references cited by the examiner, namely United States Patent No. 5,745,571 issued to Edward Andrew Zuk (hereinafter, "Zuk"), United States Patent No. 5,510, 698 issued to

Aleksandar M. Stankovic et al. (hereinafter, "Stankovic et al."), United States Patent No. 4,4,855,690 issued to Donald R. Dias (hereinafter, "Dias"), and United States Patent No. 4,800,590 issued to Vaughan (hereinafter, "Vaughan").

6. I submit this Declaration to present to the United States Patent Office, in an authenticated manner, objective facts concerning the teachings of the references and the non-obviousness of the claims relevant to the rejections made by the Examiner in the Office Action mailed 04/30/02 (hereinafter, "the Office Action").

7. I founded Quantum World Corporation in late 1993 or early 1994 to develop, manufacture, and sell the random number generator disclosed and claimed in the application.

8. By early 1994, I had made a prototype of the random number generator as claimed in claims of the application, and by November 1994 I had contracted to have quantity custom circuit boards, housings, and other parts necessary to make the random number generator manufactured in commercial quantities.

9. In November 1994, I arranged to have Mr. Carl Forest prepare the present patent application, and by February 14, 1995 the application had been prepared and filed.

10. On May 23, 1995 I sold the first random number generator as described and claimed in the application.

11. Sometime in 1995, I created a website which I have used continuously since then to advertise and sell the random number generator as claimed in the claims.

12. Dr. George Marsaglia is a well-known and respected random number expert. He is Professor Emeritus of Statistics and Supercomputing at Florida State University. Dr. Marsaglia has been instrumental in creating a testing system to rate random number generators on their effectiveness in generating true random numbers.

13. I sent one of my random number generators as described and claimed in claims 21 – 24, 27 and 28 of the present application to Dr. Marsaglia in early September 1996.

14. On September 17, 1996 Dr. Marsaglia informed me via email that my random number generator had passed all his tests, and that it was the best device he had ever tested, and, indeed, was the only device he had ever tested that passed all of his tests. See the email attached hereto as Exhibit B.

15. During the entire time from late 1993 through 1998, I was immersed in the subject of random number generators. For example, I studied all the relevant technical literature I could find while designing the random number generator and did exhaustive searches on the subject in the Denver Public Library and on the Internet before going to Mr. Forest regarding the patent application. Since that time, I have followed the literature and searched the Internet regularly for any information on the subject.

16. In 1994, the only true random number generators available were custom devices, and only a few corporations and universities had one. True random number generators were so scarce that IBM actually published random digits in a book so others could use them.

17. During the entire time from late 1993 through about 1997, I did not find any reference, product, document, Internet site or any other information that showed or suggested the interfacing of a random number generator to a computer via a device driver, a TSR, a portion of the operating system of the computer, and a program stored in the bios memory of the computer.

18. Sometime about 1997 or 1998, I became aware of other random number generators in the market.

19. One such random number generator is sold by Orion Products, having a business address at Herenmarkt 10, 1013 ED, Amsterdam, The Netherlands. See attached Exhibit C. This random number generator is as described in claims 57 – 62, 65 and 66 of the application.

20. Another such random number generator is sold by Protego Information AB, having a place of business at Citadellsvagen 11, S-211 18 Malmo, Sweden. See attached Exhibit D. This random number generator is as described in claims 57 – 62, 65 and 66 of the application.

21. There are now about six companies that sell true random number generators as disclosed in the application and claimed in one or more of claims 57 – 66.

22. On October 27, 1998 I received and filled an order from Intel Corporation, having a place of business at 2111 NE 25th, JF2-53, Hillsboro, Oregon 97124, for two random number generators as described in the patent application and covered by claims 57 – 62, 65 and 66 of the application. See Exhibit E. Later they bought another two.

23. In early 2000, Intel began shipping their Pentium III microprocessor chip set with a random number generator as part of an Intel chip including their Pentium III covered by at least claims and 57, 59, 64 and 66 of the application. Intel is expected to sell about ten million of these random number generators this year.

24. The random number generator made by Intel is very similar to the design of my random number generator, except that it is incorporated into an integrated circuit chip.

25. The random number generator made by Quantum World Corporation, Orion, Protego, the other companies that are essentially copying my invention, and Intel are all vastly superior to the software-based pseudo random number generators that were the only random number generators available for use with computers at the time of my invention of claims 57 – 66 of the application, and have essentially replaced such random number generators in the marketplace for applications requiring random numbers.

26. The above shows that my invention as claimed in claims 57 – 66 has created a new industry with sales of millions of units per year.

27. Turning now to the references cited in the Office Action, FIG. 5 of Stankovic et al. actually shows a computer which, in the words of Stankovic et al., "develops a microcode for random switching." This is one of the software-based pseudo random generators that I described in the Background of the Invention of the application.

28. As discussed in the present application on page 2, lines 18 through 27, at the time of the invention, computers are used to develop pseudo random numbers using software algorithms. Thus, Stankovic et al. does not disclose a true random number generator.

29. The random numbers generated by the random number generator of Stankovic et al. are not used by a computer, but rather are generated by the computer and used by the level shifter 52 to control the shifting of a transistor 42 (FIG. 4). See column 5, lines 19 – 27. Thus, there is no need for an interface between a random number generator and a computer.

30. Put another way, the random numbers in FIG. 5 are clearly coming <u>out</u> of the computer, not going into the computer.

31. Stankovic also discloses that the system that provides the random numbers in FIG. 5 for use by the level shifter, i.e., the computer and microprocessor, can alternatively be replaced by a hardware-based random number generator. Here again, no interface to a computer is necessary, because the random number signals go to the level shifter, not a computer.

32. Dias does not disclose a true random number generator. Dias discloses generating pseudorandom numbers using an analog oscillator 32 and a voltage controlled oscillator (VCO) 34, the combined output of which is sampled by a digital circuit See, col. 4, line 14, thorough col. 8, line 38. All of the elements described in Dias are deterministic, and as such, are inherently incapable of generating true random numbers. Dias admits this when he states that the numbers generated are "essentially" random. Col. 8, lines 39 - 43.

33. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. §1001, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

July 28, 2002
Date

Scott A. Wilber

CURRENT CLAIMS 09/699,523

57. (Amended) A method of producing a series of true random numbers, said method comprising:

using a hardware device to produce an analog noise signal;

converting said analog noise signal to a binary true random sequence of signals;

interfacing said binary true random sequence of signals to a general purpose personal computer; and

utilizing said interfaced binary true random sequence of signals in said computer.

58. A method as in claim 57 wherein said step of interfacing comprises providing a device driver.

59. (Amended) A true random number generator comprising:

a true random number generator circuit for generating a true random sequence of signals; and

a computer including a means for interfacing with said true random number generator circuit, said means for interfacing consisting of one or more of the following: a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer.

60. A true random number generator as in claim 59 wherein said means for interfacing comprises a device driver.

61. A true random number generator as in claim 59 wherein said means for interfacing comprises software for testing said true random number generator circuit.

62. A true random number generator as in claim 59 wherein said true random number generator circuit is located in a module separate from said computer, and said true random number generator further includes a cable for electrically connecting said module to said computer.

63. A true random number generator as in claim 59 wherein said true random number generator circuit is located on an add-on board for mounting in said computer.

64. A true random number generator as in claim 59 wherein said true random number generator circuit is located on the motherboard of said computer.

65. A true random number generator as in claim 59 wherein said true random

EX. A

number generator circuit is located on a peripheral of said computer.

66. A device for interfacing with a true random number generator, said device comprising:

a computer including: memory means for storing information for interfacing with a true random number generator circuit, and processing means communicating with said memory for interfacing with said true random number generator; and wherein

said information for interfacing with a true random number generator consists of one or more of the following: a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer.

EX. A

To: geo@stat.fsu.edu
From: comscire@rmii.com (Quantum World)
Subject: Re: Comscire QNG

>I have tested your QNG device---extensively---and it passes
>all tests. It is the best---indeed, the only---device I
>have tested that does so.
>
>Thank you for sending it.
>
>George Marsaglia
>

I want to thank you for your time and attention in testing our device. We are of course pleased to have corroboration (to the level we are able to test) of its statistical properties from someone who knows what to look for.

I have been searching and experimenting for years to find more powerful methods of finding patterns, or defects, in 'random' sequences. So far only serial correlation and probability of 1 (or 0) are the only tests which can be run continuously to accumulate statistics over billions --or tens of billions-- of bits.

What is required is a test which gives a running (cumulative) probability of every pattern which has occurred since the beginning of the test. Such a test is probably not achievable given any computer or algorithm I know of.

If you know of any more powerful tests, even theoretical ones, I would like very much to hear of them.

I have felt for some time that given the proper kind of test, patterns will be found in any true random sequence produced by any physical device. These patterns will arise from the interaction or relationship of our perception (or consciousness) with the underlying mechanisms involved with the production of the sequence.

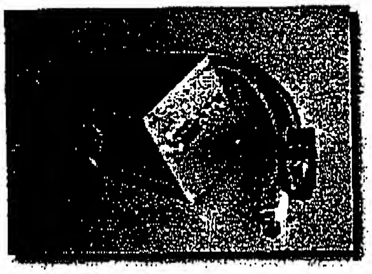
Scott Wilber
Comscire

EX. B

ORION

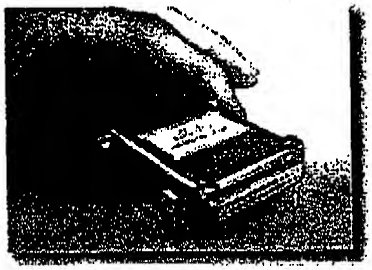
NEW!!

ROBOOT is a unique external device which continuously checks whether a server is still on-line. It will automatically reboot the machine if it crashes or freezes.



Random Number generator

The Random Number Generator card (Mac/DOS/Win) is the first true RNG to pass Marsaglia's famous DIEHARD randomness test. It produces completoly independent series of numbers and can be used for randomisation of numbers, encryption purposes, virtual casinos or for scientific research.



All Orion products are distributed and supported by ICATT. Please check the order form for prices. Mail to: tools@interact.nl

HOME HELP DESK

EX. C

ORION's Random Number Generator

- What is a Random Number Generator
- Applications of RNG's
- How the ORION RNG works
- Tests before delivery
- How to connect the RNG to the computer
- How to use the RNG in research or practical applications
- Specs of the RNG
- Software
- Prices

What is a Random Number generator

Random Number Generators are generating numbers in a sequence in such away that the next number has no relation with the previous numbers. There are software routines that may generate these numbers but these only approach the ideal of independence between the subsequent numbers because they still use an algorithm to calculate the next number from the previous one. There are several of these algortihms and it is difficult, though in principle not impossible, to assess from the numbers which algorithm is used.

Another approach to obtain random numbers is to use noise in nature. The most commonly used noise sources are bouncing of ping pong balls (in lotteries) or more sophisticated: radioactive decay or electron tunneling in electronic components. These processes are in principle unpredictable. ORION's Random Number Generator is of the latter kind.

Applications of the RNG

RNG's are applied in cases where one needs unpredictable events. Theyare used in scientific research eg. in so-called Monte Carlo simulations. Theyare also used in research on so called paranomral or psi phenomena whererandomness is an essential condition and they are used in encryption/decryptionapplications.

How the RNG works

ORION's Random Number Generator consists of two independent analogue Zener diode based noise sources. Both signals are converted into random bitstreams, combined and subsequently transmitted in the form of bytes to the RS-232 port of your computer. Special timing circuits ensure that crucial logical operations occur at moments that the device has stable signals.

The baud rate is 9600. So the device is capable of supplying you with about 960random bytes or 7600 random bits per second

Power is drawn from the RTS and TXD signal. (pins 4 and 2 of the D-25connector). In order to work properly the RTS signal should be high (5 volts orhigher) and one should not send bytes to the device!

WARNING: part of the RNG is shielded. Do not open the device. It is not allowed to copy or use the design of the RNG without written permission of the original developer, the Foundation for Fundamental Researchon Man and Matter (FREMM).

The testdata

Each RNG passes a 256 run random ness test before being shipped. Each run consists of 8192 8-bit samples. The unselected results of this test are included with the package in the form of the number of bits that have been used, the over-all first bias, and an estimate of the higher order biases. Detailed test can be obtained upon request. If the first order bias is larger than1 bit in 2000 bits the RNG will be rejected. Please read the important NOTE on the next page on how to compensate further for remaining first orderbias.

How to connect the RNG to the computer

The RNG is functionally similar to a 'crazy' modem sending random information to your machine. It should be connected like any modem. In some cases when the computer has a standard D-25 male connector you can plug the RNG directly into that connector. In other cases (e.g. for Apple computers) you have to connect the RNG through a standard modem cable.

How to use the RNG in research or practical applications.

The byte read from the RS-232 port should be seen as a number. The lowest value is 0 and the largest value is 255. In experiments where a binary decision is needed you will have to transform the byte (0,255) into a '0' or a'1'.

For instance:

```
r= 0

byte = read_com()

IF (byte > 127.5) then r=1
```

For decisions between 4, 8, 16 etc. alternatives a similar procedure can beused. If one needs to have 5 alternatives (for instance if you want to simulate a Zenercard experiment) then the procedure may be as follows:

```
start: byte=read_com()
```

```
if byte > 249 then goto start

r = int (byte/5)
```

IMPORTANT NOTE

Although the design is such that the device has a first order bias whichis guaranteed smaller than the specification we strongly recommend the following *guidelines* when using the RNG in any crucial application.

1. Have a short straight test run of the RNG at the beginning and at the end ofthe sessions, just to check for first order effects. A few seconds sampling is sufficient. Look at the supplied test programs as an example .

2. Before using the random byte as read from the RS 232 port, apply a software routine that ensures that in 50% of the samples a logical '1' will be interpreted as '0' and of course also that in 50% of the samples a logical '0'will be interpreted asa '1'.This will ensure that no systematic first order deviation will arise even when the device does not function properly (of course you might get strong second order and variance effects). The easiest way to do this is to transform a '1' into a'0' and a '1' in a '0' on odd trials.

A more sophisticated and recommended way is to XOR the random byte with a pseudo random byte. In that case the resulting bytes will even behave properly for higher order bias effects.

3. **In case of psi research:** Always have a no-subject condition as part of theformal design. In the no subject condition each button-press is replaced by a random wait routine. As a rule of thumb run the no-subject condition about 10 times more often than the experimental subject condition.

Due to the limited knowledge we have of the psi phenomenon we cannot specify a control condition which is guaranteed psi-free. There fore one might occasionally also find deviations in the no-subject condition. Just report these.

Specs

Requirements: *Works with any DOS, Windows machine and Macintosh*

Required power: *None*

Dimensions: *appr. 15 * 30 * 40 mm*

Biases: *First order less than 1 in 4000; Higer order less than 1 in 14000000.*

DataRate: *9600 Baud*

Communication Protocol: *RS232*

Software

There is software available for the Macintosh and for DOS or Windows systems. The software can be downloaded here.

Description of software for DOS/Windows and Macintosh

Download software for DOS & Windows

Download software for Macintosh

The sources are available on request. Listings are already available.

Prices

Please check the order page for prices and information on ordering.

ORION's Random Number Generator

Description of SOFTWARE

RNG-software for DOS and Windows

RTEST <port> <ncycles>

<port> indicates which serial port is used (either 1 or 2. (default:2)).

<ncycles> gives the number of test runs. One test run consists of reading 26500 bytes and calculating the Chi-2 and the number of bits on each one of the eight positions within the byte. If more test runs are specified the numbers (except the chi-2) accumulate.

If the program is started it will display one sample from the specified port. More samples can be displayed by hitting the ENTER-key.

Continuous sampling of the test runs is started by the ESC key. The results will be available for further statistical analysis in the textfile 'TEST'.

RTEST is also available for Windows.

[Download RNG software for DOS/Windows]

RAND3 takes one argument.

RAND3 <port>

This program starts in the same way as RTEST by displaying individual samples after each time the ENTER key is hit. . After hitting the ESC key continuous sampling starts and the bytes are displayed tin a graphical form in real time. The continuous samples can be restarted by hitting the space-bar. Exit through the ESC key.

The software addresses the serial port directly. It is also possible to implement a driver which is interrupt driven (like the software supplied for the Mac). A driver is available upon request.

RNG Software for Macintosh

The Macintosh software is based upon the standard serial port drivers rather than on direct access of the serial ports. The bufferspace for this driver should be large enough to hold incoming bytes while the driver is active and no reading occurs. In cases where this is not clear one should close the driver and reopen it when needed.

In the examples the modem-port is used. In the listings it is indicated how to use the printer-port.

RNG_test

The program will ask for the number of test runs to perform. Each test run consists of 8192 samples. The chi-2 (df=255) and the number of '1' bits minus the number of '0' bits on each of the positions within the byte are displayed after each run. Chance expectation is of course 0. When more runs are specified the cumulative number of '1' bits minus the cumulative number of '0' bits are displayed as well as the corresponding standard deviation.

All data are saved in the file RNGTESTDATA. Each run on a separate line and the data separated by spaces.

[Download RNG software for Macintosh]

PK_test

This program asks for the number of runs, the number of trials per run and a parameter which will enable you to set the trial-rate. You specify also an *id* which will be the same as the filename where the results are stored.

[Download RNG software for Macintosh]

Writing your own software

DOS

The DOS disk also contains the files RTESTR.cpp and RAND3.cpp. These are the C++ source codes for the supplied test programs. It is straightforward to translate the functions

init_com, com_ready and read_com into any other accepted programming language.

The program uses one address to indicate where the port in the address space is located. The symbolic name is: BASE. For all DOS-computers this is a standard (03F8hex and 02F8hex for PORT1 and 2 resp.). If you want to use the RNG with a non standard DOS machine it could be that the serial ports are differently located and the internal registers are differently used.

In that case you have to write your own code for the routines:

init_com (initialising the port on 9600 baud, 8 bits, no parity and 1 stopbit)

com_ready (returns '1' if the port has read a byte, '0' when nothing happened)

read_com (returns the byte from the port)

The init_com routine also sets the RTS line high. So even if you use a batchfile or a system command to set the proper baudrate etc. you might need to set this RTS.

In *BASIC* one could use a POKE statement (but be careful to define the proper segment with DEF SEG; consult your manual)

```
rem init_com for port 1

BASE_ADDRESS= 0X03F8

POKE BASE_ADDRESS+4, 2

rem com_ready; status = 1 means there is a byte available

IF (int(peek(BASE_ADRESS +5)&& 1) THEN status=1 ELSE
status=0

rem read_comm

byte= peek(BASE_ADDRESS)
```

Macintosh

The Mac keeps the RTS line after opening the serial driver high thus ensuring proper functioning of the RNG.

ORION Prices & Order form

ROBOOT and RNG are distributed by ICATT.
Please print this form, fill it in, sign it and **Fax it to ICATT at + 31 20 4206075** or send it with a cheque.

Name:	....	Company/ Institution:	....
Address:	....	Zipcode:	....
City+State/Province:	....	Country:	....
E-mail:	....	Tel:	....
	(please print clearly)	Fax:	....

I wish to order:

○ 1 ROBOOT: **CURRENTLY UNAVAILABLE ($ 225)***
○ 1 ROBOOT *educational*: **CURRENTLY UNAVAILABLE ($ 195)**
○ 4 ROBOOTs or more: **10 % discount.**
○ 15 ROBOOTs or more: **20 % discount.**
Number:...

○ 1 RNG: 1152 Dutch Guilders **($ 588)***
○ 1 RNG *educational*: 576 Dutch Guilders **($ 294)**

○ Shipping in Europe: 19.60 Dutch Guilders **($ 10)** for each ROBOOT/RNG shipped.
○ Shipping outside of Europe: 29.40 Dutch Guilders **($ 15)** for each ROBOOT/RNG shipped.

Residents of the European Community: please add 17.5% BTW (VAT)

Total:...

Payment: ○ Mastercard ○ Visa ○ Check (send with this form)

Credit card number: ....

Expiration date: ....

Any comments: ....
....

Signature: ....

Fax this form to ICATT +31 20 4206075
or send it with a cheque to ICATT - Herenmarkt 10, 1013 ED, Amsterdam, The Netherlands

* Please send us cheques quoting Dutch Guilders. US Dollar prices can

vary slightly depending on exchange rates.

About

About Protego Information AB

Protego Information AB is a privately held company with its development centre located in Malmoe, Sweden. Protego is a company dedicated to the development and marketing of computer related security products. Securing digitally stored information in storage and in transfer is our business. We develop encryption algorithms and protocols for use in our own encryption products. We are continuously developing security technology for securing communications on the Internet and on Intranets. We also develop encryption products for securing stored information on personal computers. In house designed hardware for cryptographic use is a technology that is an integrated part of many of our products in development.
Our motto is "encryption should be easy to use but hard to break".

Protego Information AB
Citadellsvägen 11
S-211 18 Malmö
Tel. +46 (0)40 94 05 00
Fax. +46 (0)40 30 36 46
E-mail to postmaster@protego.se

About Technology Products Contact Order Home

EX. D

Technology PROTEGO INFORMATION

General Design Considerations

The SG100 generator divides into a hardware noise generator and a software driver. Between the SG100 hardware and the SG100 software driver is a computer port with a port driver. The important question here is how to cleverly split our mission into two parts: one suitable for processing in the hardware and the other suitable for the software driver; and how to overcome possible problems with the computer port in the middle. These questions affects throughput, quality, and reliability.

In the SG100 generator we have chosen to do as little processing as possible in hardware. This results in poor statistical performance, when measured on the raw hardware, but actually gives us several advantages.

First: Building hardware is expensive, at least if we compare to a software solution of the same problem.

Second: If no processing of the noise takes place in hardware it is simple to calculate a figure just how well the device is operating. Not only do we have the opportunity to install run-rime testing of the device, we can actually remedy a situation with a low noise output, if we know when it occurs. This makes the SG100 generator more reliable than other similar devices which lack run-time testing.

Third: The compatibility is increased. SG100 works on almost any serial port, facilitating the use of the same hardware on all platforms.

Hardware Properties

Hardware connectable to any computer.
Powered from the computer port - no batteries or cables.
Device automatically switches off on battery-powered computers.
High resistance against power fluctuations.
High resistance against external electromagnetic fields.
Information feed into computer difficult to intercept.
Runtime electrical and statistical testing.
High output speed: up to 9.500 bytes/sec.
Can be manufactured in a reliable way.
No time-consuming factory adjustments.
Pass CE requirements.

Design Considerations for Hardware.

We have chosen to manufacture the SG100 generator for 9-pin serial ports. Even if the PC-parallel/printer port provides higher communication speeds there could be problems when writing drivers for Win32 or UNIX-systems. As the noise process generates a serial stream of information any possible benefit of a parallel interface is clearly limited.

Only very little power can be obtained from a serial port. This has been a major construction problem, but now the device operates well even far out of the specification of the serial port.
We have found, experimentally, that a number of PC-models have serial ports which operates below the RS-232C standard. The SG100 driver monitors power and if any power failure is detected this error is forwarded to the calling application.

The hardware has a power-off mode as well as an operating mode. When no more noise is needed the software driver will switch the SG100 to power-off mode. If your portable computer has automatic power saving (APM-hardware) this will work too.

Resistance against Radio Frequency (RF) fields has been in the specification from the beginning of our product development cycle. No customer has to fear that the operation of the SG100 generator can be intentionally influenced from any external RF-field. This has been accomplished by using a noise generating process with a high output level and by using a RF-shield casing.
The SG100 generator has a built in RF-field filter.

The reader should note that all ordinary computers are somewhat sensitive to strong RF-fields.
We estimate that almost all computer models will cease to operate when subject to radiation
of a strength high enough to influence the SG100 generator.

The SG100 has a plastic casing which also protects against RF-fields. In the future we will also
provide a metal casing for demanding customers with EMP/HEMP/TEMPEST protected
computers and high security installations. Measurements of the actual levels of fields involved
in this discussion will be published here, when available.

For a serial port the baudrate of the port must be specified. To overcome the fact that some
computers don't support all baudrates, and to avoid specifying a low baudrate (like 9,600),
we have choosen not to convert the output from the SG100 generator into byte-serial form
before feeding it to the computer. The hardware outputs an irregular square wave that is feed
into the computer UART which samples and converts the input stream into digital form. During
this process the computer UART selects which parts of the noise stream that will be interpreted
as start bits and which parts that will be ignored as stop bits. The decision of when to interpret
the stream as "1" or "0" is up to the UART. You may use any baudrate below 100,000 that is
accepted by the serial interface driver.

Not converting the SG100 output into byte-serial form simplifies hardware design and makes
interception of the serial stream very difficult, as it is unknown what bits are selected as startbits
and as the sampling in the UART is somewhat different on all computers.

Software Properties

Windows-95 and Windows NT driver delivered with product.
Easy to use API interface.
Immediate action if the device fails.
Fast response to the calling process.
Interface for multiple processes reading noise.
No cryptographic or statistical weaknesses.
Do not deliver low quality noise when first called or if called repeatedly.
Easy to include drivers/etc in OEM product.
Source code access for OEM-customers.

Drivers can be written for any platform.

Design Considerations for the Software Driver

A driver for Windows 95 and Windows NT is delivered with the SG100 noise generator. This is the same driver that is used by Protego InfoSafe line of security products. The API of the driver (*.h file) is included as well as compiled demo programs in C/C++. Using the provided compiled demo program you may extract noise to a named file to facilitate an easy interface with almost any statistical or security product.

The driver may be linked directly into the EXE of an OEM product. If the customer wish to modify the driver himself the source code will be made available on a non-disclosure basis.
It is also possible to use the SG100 generator on other (non PC) platforms. In that case a request can be made for the complete driver specifications and algorithms.

The provided SG100 driver DLL operates the hardware and checks for hardware errors. A wide range of hardware errors can be detected and action taken accordingly. If the device fails any pending call for noise will be returned with an error status. If the user disconnects the SG100 generator a signal is given to the calling application to let the user reinstall the noise generator without any influence on the running application.

The driver do a continuous statistical check of the hardware. The driver then calculates the rate of information obtained form the SG100 generator. If the rate falls below 70% the device is considered faulty. A rate above 96% is acceptable without further action and if a rate lower than 96% is obtained this will be remedied by reading the noise stream twice, giving the driver access to the double amount of input. Reading twice, if necessary, will give us only half overall throughput.

The hardware, built according to a minimal noise-processing criteria, cannot create complex output correlations by itself. We will have a statistical bit bias from the hardware, with "1" and "0" not occurring with exactly 50%-50% distribution, and also a correlation between adjacent bits (and a correlation between the start bit and the first bit). We see that there cannot be any high complex correlations because the hardware lacks memory. After a short period of time a previous noise stream cannot influence the current noise stream.

Suppose we read four bytes and form a 32 bit integer A. Suppose we know that the current information flow is 94%. We then read a second 32-bit integer B, at some other point in time, maybe a few seconds later. We now know that A and B are independent and random, and also that we have statistical deficiencies, as indicated above. To form a 32-bit integer C with almost 100% information rate it is sufficient to add A and B using 32-bit binary add (with carry) C=A+B. We may picture this by an arrow A pointing on the circumference of a circle divided into 2^32 segments. The selection of A has some bias, making not all the 2^32 segments equally probable. If we now turn the arrow B steps forward it is visualised that if the bits of A and B has the independence property, above, the number C will have surprisingly good statistical properties. Note that 2*94%=192%; the number C, consisting of 32 bits, clearly cannot store any more information than 32 bits (100%).

In this way, using a similar but more complex "adding", the driver can guarantee that the minimum information flow is 96%. Using a SG100 developer package the reader may obtain the raw SG100 output and verify this for himself.

The software driver has a buffer of noise ready for reading to facilitate fast response when called. Currently this has been set to 32.000 bytes, as no more is necessary for any of the InfoSafe products. The buffer also has an additional purpose, here is the pre-processed input from the driver "whitened" to enable the output stream to pass any statistical or cryptographic test. This is done using a combination of statistical and cryptographic techniques.

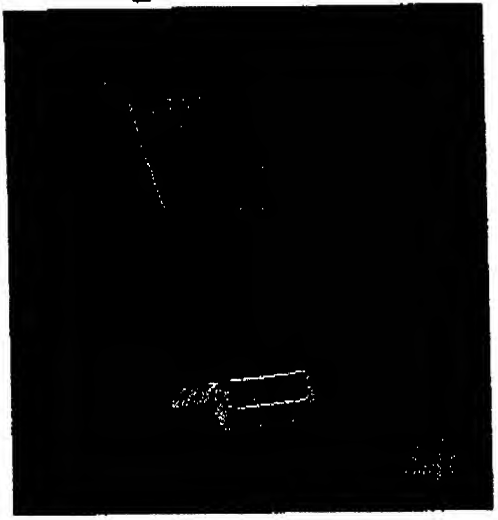
This relieve the customer, almost regardless of application, from the need to further improve the SG100 driver output. The detailed cryptographic and statistical methods can be obtained by customers of the SG100 development package.

The software driver uses process synchronisation allowing multiple processes to read noise simultaneously. The synchronisation also blocks processes if the noise buffer becomes empty or during an initial start-up period before the noise buffer has been thoroughly "whitened".

When the noise buffer becomes empty the driver reads tree times the buffer size to allow filling the buffer with a maximum information content. We have seen that the minimal information input rate is 96%. The reader should note that it is possible to read this minimum information content only if an application asks for a very long continuous string of noise, and that the 32.000 first bytes of that long string will most probably be of 100% quality. To experimentally obtain the (maximum) 4% lack of information, the output must be intentionally processed to break the cryptographic operations, which will be most difficult mainly because of the small statistical deficiency searched for. This is practically infeasible for the string lengths that the drivers has been designed for.

Some demanding customers will not accept even the most diminutive statistical deficiency. Solving this problem will, in addition to other steps, including a halving of output throughput, forcing the information content to 100% at all times. If asked for, a double SG100 interface can be made available.

Protego Information SG100 Data Sheet



No more unreliable generation of crypto keys and random numbers in software, the viable alternative is here! SG100 Security Generator is an easy to use easy to integrate hardware random number generator that connects to the standard serial port. Complete with driver software for Windows 95 / Windows NT and example programs in source.
SG100 is the choice if you want to strengthen and enhance your encryption, statistics and simulation software.
If you want to OEM the SG100 contact us for further information and prices.

SG100 is delivered with driver for Windows-95 andWindows NT. The device is connected to the computer through the 9-pin serial port. Power is taken from the port. Supports all baud rates up to 100,000 baud. Throughput is about 9,500 bytes/sec for the 100,000-baud rate. Output is processed using statistical and cryptographic methods, and passes any statistical test. Resistant to external electromagnetic fields and high resistance against power fluctuations.

- Windows-95 and Windows-NT driver delivered with product.
- High output speed: 9.500 bytes/sec.
- Device powered from the computer port - no batteries or cables.
- Runtime electrical and statistical testing.
- Easy to include drivers in OEM product.
- Fast response to the calling process
- Interface for multiple processes reading noise
- Information feed into computer difficult to intercept.
- No cryptographic or statistical weaknesses.
- Driver can be written for any platform.
- Pass the Diehard test

Click HERE for technical data on the SG100 Security Generator.

Prices and availability.

The SG100 is available in two basic packages; Developer and Runtime.

Both SG100 Developer Packages and SG100 Runtime packages are available now.

The Developer package contains:

- One SG100 Generator.
- Drivers for Windows 95 and Windows NT in DLL format with C/C++ header file.

- Demo Programs, compiled to EXE including C/C++ source, that open and use the driver DLLs.
- Hardware Test Programs (EXE only) for the SG100 hardware.

Price : USD 140
[Go to Order page to place order]

Runtime Packages
No drivers are included (you may copy drivers/etc from a Developer Package).
Note! You need to be in possession of one Developer Package and a general license, included in the
Developer Package, to copy and use the driver software for the SG100 generator.

Quantity	Price in USD/unit
10	103
50	93
100	85
+300	Contact Us!

Prices and specification subject to change without prior notice.

About Technology Products Contact

COMSCIRE
The Quantum World Corporation
P.O. Box 1930
Boulder, CO 80306-1930
(303) 443-2420
(303) 443-2450 Fax

PACKING LIST

INVOICE NUMBER:	**102798-1**
BILL TOTAL:	**$636.00**
DATE:	**Oct. 27, 1998**
TERMS:	**prepaid VISA**
PO NUMBER:	**verbal**

DELIVERY TO:	Debbie Cabanas
ADDRESS:	Intel Corporation 2111 NE 25th, JF2-53 Hillsboro, OR 97124
TELEPHONE:	(503)264-8396
SHIP DATE:	Oct. 27, 1998
NOTES:	

BILL TO:	prepaid
PURCHASE REP.:	Debbie Cabanas
ADDRESS:	same as above
TELEPHONE:	
NOTES:	

ORDER DESCRIPTION

QUANTITY:	PRODUCT DESCRIPTION:	PRICE (EACH):	SER. #:	NOTES:
2	QNG Device and Software	$295.00		NT Ver1.2
0	A/B Switchbox*			
0	Parallel Port Card*			
0	Extra Cable*			
	PRODUCT TOTAL:	$590.00		

SHIPPING

DESCRIPTION:		COST	NOTES:
2-Day			
Overnight	XXXXXX	$46.00	
Other			

This is the shipping charge for one QNG Device. Shipping charge for additional QNG Devices or options accessories will vary.
*Optional Accessories
In stock items will normally ship within one working day.

EX. E

PATTON BOGGS LLP
ATTORNEYS AT LAW

867 Coal Creek Circle, Suite 200
Louisville, CO 80027-9750
303-379-1100

Facsimile 303-379-1155

To:	**Examiner Tan V. Mai**
Company:	Commissioner Of Patents And Trademarks
Fax Number:	1-703-746-7239
Phone Number:	1-703-305-9761
Application No.:	09/699,523
Total Pages Including Cover:	**40**
From:	**Elaine C. VonSpreckelsen, Secretary to Carl A. Forest**
Sender's Direct Line:	303-379-1111
Date:	July 29, 2002
Client Number:	13323.102 (Formerly 2022/002D1)

Comments:

Serial No.:	09/699,523	Group No.	2121
Filed:	30 Oct 2000	Examiner:	Mai, Tan V.

Attached please find the following documents responsive to the Office Action mailed April 30, 2002 to be entered in the above application:

1. Amendment and Remarks (11 pages)
2. Declaration Of Scott A. Wilber, with attachments (28 pages)

Thank you for your assistance in this matter.

ANCHORAGE
BOULDER
DALLAS
DENVER
NORTHERN VIRGINIA
WASHINGTON, D.C.

Confidentiality Note: The documents accompanying this facsimile contain information from the law firm of Patton Boggs LLP which is confidential and/or privileged. The information is intended only for the use of the individual or entity named on this transmission sheet. If you are not the intended recipient, you are hereby notified that any disclosure, copying, distribution or the taking of any action in reliance on the contents of this facsimile is strictly prohibited, and that the documents should be returned to this Firm immediately. If you have received this facsimile in error, please notify us by telephone immediately so that we can arrange for the return of the original documents to us at no cost to you.

If you did not receive all of the pages or find that they are illegible, please call 303-379-1111.

#10/D
ME
8-6-2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

U.S. Patent Application No.: 09/699,523	Group Art Unit: 2124
Filing Date: October 30, 2000	Examiner: Mai, Tan V.
For: Random Number Generator And Generation Method	Docket No: 13323.102 (Formerly 2022/002D1)
Applicants: Scott A. Wilber	

CERTIFICATE OF TRANSMISSION UNDER 37 CFR 1.8

I hereby certify that this correspondence, along with all papers referred to as being transmitted, are being facsimile transmitted to the Patent and Trademark Office Fax No. (703) 746-7239.

Date

Elaine C. VonSpreckelsen

BOX NON-FEE AMENDMENT
ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, DC 20231

Dear Sir:

This Amendment and Remarks are responsive to the Office Action mailed April 30, 2002. A Declaration of Scott A. Wilber, hereinafter referred to as "the Declaration", is enclosed.

AMENDMENT

In the Claims:

Please amend claims 57 and 59 as follows:

57. (Amended) A method of producing a series of true random numbers, said method comprising:

using a hardware device to produce an analog noise signal;

converting said analog noise signal to a binary true random sequence of signals;

interfacing said binary true random sequence of signals to a general purpose personal computer; and

utilizing said interfaced binary true random sequence of signals in said computer.

59. (Amended) A true random number generator comprising:

a true random number generator circuit for generating a true random sequence of signals; and

a computer including a means for interfacing with said true random number generator circuit, said means for interfacing consisting of one or more of the following: a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer.

REMARKS

A. The Office Action Has Not Established A Prima Facie Case of Obviousness

Two amendments have been made to the claims. In considering some of the examiner's arguments, the undersigned realized that a key element of the method claim 57 was expressed in terms of an apparatus limitation, i.e., "hardware". While this is not contrary to patent law, it could lead to some lack of clarity; therefore, the claim was amended to express this limitation more fully in terms of processes. The amendment is supported in the specification at page 4, line 13 – page 5, line 16, page 7, line 25 – page 8, line 8, page 14, last line – page 15, first line, and elsewhere. It was also noted that, while the word "true" was used in the first part of the second line of claim 59, it was not used in the last part, and thus there could be some ambiguity. Therefore, this was corrected also.

Claims 57 – 66 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Zuk. This rejection is respectively traversed.

Zuk discloses a system and method for securing communication between a smart card 6 and a key generation center 4 which includes a processing system 8. To facilitate this, the smart card includes software routine C1 (column 4, lines 12 – 15) which generates a single random number (column 5, lines 1 – 4) that is passed to the key generation center via the smart card communications interface 20 (column 5, lines 5 and 6, and column 3, line 33). After the single random number is generated, the routine C1 that produced it is erased (column 5, lines 27 – 31). Zuk does not include three limitations of claim 57: a) it does not use a hardware device to generate an analog noise signal – it uses a software routine to generate a random number; b) it does not generate a sequence of true random signals – it generates a single random number; and c) it does not interface the signals (plural) to a general purpose computer and use the signals (plural) in the computer.

All of the limitations of a claim must be considered when weighing the differences between the clamed invention and the prior art in determining the obviousness of a method claim. MPEP 2116.01 and 2143.03.

To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all of the claim limitations.

A prima facie case of obviousness of claim 57 cannot be made with Zuk, because it does not contain three essential limitations in the claim. There is no motivation in Zuk to interface a sequence of random signals to a general purpose computer because the only purpose of the random number in Zuk is to facilitate a single secure communication. Thus, Zuk cannot form a basis under 35 U.S.C. 103(a) to find claim 57 obvious.

Similarly, with regard to claim 59, Zuk does not contain two essential limitations: a) it does not generate a sequence of signals, and b) it does not disclose or suggest a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer to interface to the computer. There is no motivation in Zuk to interface a sequence of random signals to a general purpose computer using a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer, because the only purpose of the random number in Zuk is to facilitate a single secure communication. Thus, Zuk cannot form a basis under 35 U.S.C. 103(a) to find claim 59 obvious.

Likewise, with respect to claim 66, Zuk does not disclose or suggest a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer to interface to the computer. Further, there is no motivation in Zuk to interface to a general purpose computer using a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer, because the only purpose of the random number in Zuk is to facilitate a secure communication with a smart card. Thus, Zuk cannot form a basis under 35 U.S.C. 103(a) to find claim 66 obvious.

With regard to independent claims 58 and 60 – 65, the position in the Office Action that the detail features are obvious to one of ordinary skill in the art is respectfully traversed. There is no evidence that someone skilled in the random number generator art, prior to this disclosure, would have any familiarity with these details. Moreover, the motivation to combine these details with a random number generator is not found in the Zuk reference, but in the present disclosure. Thus, Zuk cannot form a basis under 35 U.S.C. 103(a) to find claims 58 and 60 – 65 obvious.

Claims 57 – 66 were rejected in the Office Action under 35 U.S.C. 103(a) as being unpatentable over Stankovic et al. The examiner also puts in parentheses, "Applicant's admission Prior Art", though no further reference is made to "Applicant's admission Prior Art", so it is impossible to determine what this means. This rejection is respectfully traversed.

Stankovic et al. discloses a device that uses random numbers to control random switching in a power converter (column 1, lines 7 and 8). FIG. 5 of Stankovic et al. actually shows a computer which, in the words of Stankovic et al., "develops a microcode for random switching." This is nothing more than the software-based pseudo random generators described in the Background of the Invention of the application, and inherently generates pseudorandom numbers, not true random numbers. See the Declaration, paragraphs 27 and 28. The random numbers generated by the random number generator of Stankovic et al. are not used by a computer, but rather are generated by the computer and used by the level shifter 52 to control the shifting of a transistor 42 (FIG. 4). See column 5, lines 19 – 27. Stankovic et al. also discloses that the system that provides the random numbers in FIG. 5 for use by the level shifter, i.e., the computer and microprocessor, can alternatively be replaced by a hardware-based random number generator. In this embodiment of Stankovic et al., there would be no computer, and, of course, no interface to a computer.

A prima facie case of obviousness of claim 57 cannot be made with either embodiment of Stankovic et al. because either embodiment does not disclose the following three limitations in claim 57: a) converting an analog noise signal to a binary true random sequence of signals; b) interfacing a binary true random sequence of signals to a general purpose personal computer; and c) utilizing the interfaced binary true random sequence of signals in the computer. Further, there is no motivation in Stankovic et al. to interface a sequence of random signals to a general purpose computer, because in one embodiment the computer produces the signals, and in the other embodiment, there is no computer. Thus, Stankovic et al. cannot form a basis under 35 U.S.C. 103(a) to find claim 57 obvious.

Similarly, with regard to claim 59, Stankovic et al. does not contain two essential limitations. With respect to the embodiment of FIG. 5: a) it does not include a true random number generator circuit for generating a true random sequence of signals; and b) it does not disclose or suggest a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer to interface to the computer. With respect to the second embodiment where the computer is replaced by a hardware random number generator: a) it does not include a computer; and b) it does not disclose or suggest a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer to interface to the computer. Further, there is no motivation in Stankovic et al. to interface a sequence of random signals to a general purpose computer using a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer, because in the only embodiment of Stankovic et al. that includes a computer, the random number in Stankovic et al. is generated by a computer and passed to a level shifter. Thus, Stankovic et al. cannot form a basis under 35 U.S.C. 103(a) to find claim 59 obvious.

Likewise, with respect to claim 66, Stankovic et al. does not disclose or suggest a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer to interface to the computer. Further, there is no motivation in Stankovic et al. to interface to a general purpose computer using a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer, because in the only embodiment of Stankovic et al. that includes a computer, the random number in Stankovic et al. is generated by a computer and passed to a level shifter. Thus, Stankovic et al. cannot form a basis under 35 U.S.C. 103(a) to find claim 66 obvious.

With regard to independent claims 58 and 60 – 65, the position in the Office Action that the detail features are obvious to one of ordinary skill in the art is respectfully traversed. There is no evidence that someone skilled in the random number generator art, prior to this disclosure, would have any familiarity with these details. Moreover, the motivation to combine these details with a random number generator is not found in the Stankovic et al. reference, but in the present disclosure. Thus, Stankovic et al. cannot form a basis under 35 U.S.C. 103(a) to find claims 58 and 60 – 65 obvious.

Claims 57 – 66 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Dias. This rejection is respectively traversed. Dias does not disclose a true random number generator. See the Declaration, paragraph 32. Dias discloses generating pseudorandom numbers in pseudorandom number generator 12 using an analog oscillator 32 and a voltage controlled oscillator (VCO) 34, the combined output of which is sampled by a digital circuit See column 4, line 14, through column 8, line 38. Dias admits that the numbers generated by random number generator 12 are not true random numbers when he states that the numbers generated are "essentially" random. See column 8, lines 39 – 43. The generation of the random numbers is controlled by a computer 14. See column 3, lines 10 – 47. When produced, the numbers are used in a key ring 18. See column 2, lines 31 – 35.

A prima facie case of obviousness of claim 57 cannot be made with Dias, because Dias does not contain four essential limitations in the claim: a) using a hardware device to produce an analog noise signal; b) converting the analog noise signal to a binary true random sequence of signals; c) interfacing the binary true random sequence of signals to a general purpose personal computer; and d) utilizing the interfaced binary true random sequence of signals in the computer. Further, the motivation in Dias is to not use a true random number generator, because Dias says these are too large. See column 1, lines 22 – 35. Dias teaches the use of the numbers in an electronic key ring, and says that in such key rings, there is only a small amount of chip area, and therefore the true random number generators are not desirable in this application. Ibid. Thus, Dias cannot form a basis under 35 U.S.C. 103(a) to find claim 57 obvious.

Similarly, with regard to claim 59, Dias does not contain two essential limitations: a) a true random number generator circuit for generating a true random sequence of signals; and b) a means for interfacing to a computer consisting of one or more of the following: a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of the computer. There is no motivation in Dias even to utilize a true random number generator, because Dias teaches that these are too large for the applications contemplated. Thus, Dias cannot form a basis under 35 U.S.C. 103(a) to find claim 59 obvious.

Likewise, with respect to claim 66, Dias does not disclose or suggest: a) the true random number generator aspect mentioned several times; and b) a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer to interface to the computer. Further, there is no motivation in Dias to interface to a general purpose computer using a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer, because the whole thrust of Dias is to use the system in a small device, such as a key ring. Further, as mentioned above, Dias teaches against the true random nature of the claim. MPEP 2145 X.D. Thus, Dias cannot form a basis under 35 U.S.C. 103(a) to find claim 66 obvious.

With regard to independent claims 58 and 60 – 65, the position in the Office Action that the detail features are obvious to one of ordinary skill in the art is respectfully traversed. There is no evidence that someone skilled in the random number generator art, prior to this disclosure, would have any familiarity with these details. Moreover, the motivation to combine these details with a random number generator is not found in the Dias reference, but in the present disclosure. Thus, Dias cannot form a basis under 35 U.S.C. 103(a) to find claims 58 and 60 – 65 obvious.

Claims 57 – 66 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Vaughan. This rejection is respectively traversed. Vaughan discloses security system 50 that is used to control access to a host computer 52. See column 5, lines 24 – 26. The system uses a true random number generator 80 and a microprocessor 74 to generate a pseudo-random number that is passed to a password generator 10 and stored in the EEPROM 90 in the security system 50. See column 4, lines 49 – 59. The common pseudo-random number permits the password generator to be used to enable the security lock to obtain access to the host computer. See column 3, lines 44 – 54. Vaughan nowhere discloses passing either the true random number or the pseudorandom number to the computer 52. Neither are the true random number nor the pseudo-random number ever passed through either of the RS232 interfaces 64, 65 mentioned by the examiner.

A prima facie case of obviousness of claim 57 cannot be made with Vaughan, because it does not contain three essential limitations in the claim: a) converting the analog noise signal to a binary true random sequence of signals; b) interfacing the binary true random sequence of signals to a general purpose personal computer; and c) utilizing the interfaced binary true random sequence of signals in the computer. Further, there is no motivation in Vaughan to interface a sequence of random signals to a general purpose computer, because the only purpose of the random number in Vaughan is to facilitate unlocking the lock 50. Thus, Vaughan cannot form a basis under 35 U.S.C. 103(a) to find claim 57 obvious.

Similarly, with regard to claim 59, Vaughan does not contain two essential limitations: a) a means for interfacing a computer to a true random number generator circuit; and b) a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer to interface to the computer. There is no motivation in Vaughan to interface a sequence of random signals to a general purpose computer using a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer, because the number generated is never passed to the computer. Thus, Vaughan cannot form a basis under 35 U.S.C. 103(a) to find claim 59 obvious.

Likewise, with respect to claim 66, Vaughan does not disclose or suggest a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer to interface to the computer. Further, there is no motivation in Vaughan to interface a sequence of random signals to a general purpose computer using a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer, because neither the true random number nor the pseudorandom number are passed to the computer. Thus, Vaughan cannot form a basis under 35 U.S.C. 103(a) to find claim 66 obvious.

With regard to independent claims 58 and 60 – 65, the position in the Office Action that the detail features are obvious to one of ordinary skill in the art is respectfully traversed. There is no evidence that someone skilled in the random number generator art, prior to this disclosure, would have any familiarity with these details. Moreover, the motivation to combine these details with a random number generator is not found in the Vaughan reference, but in the present disclosure. Thus, Vaughan cannot form a basis under 35 U.S.C. 103(a) to find claims 58 and 60 – 65 obvious.

B. Objective Evidence of Nonobviousness

Objective evidence that a claimed invention is nonobvious, such as long felt but unsolved needs, commercial success, and copying by others must be considered by the USPTO. MPEP 2141 and MPEP 716.01(a). The examiner is invited to carefully read the application from page 1, line 8 through page 3, line 30. This discussion, which is fully supported by references provided in the IDS, shows that there was a long felt need for the true random number generator claimed. Paragraphs 12 – 14 of the Declaration show that the invention has met that need, and is so unexpectedly superior to the prior art that the leading expert in random number generators has stated that it is the only device ever tested to pass all his randomness tests. This evidence is highly reliable because it is by a noted expert completely independent of the inventor. Further, the Declaration shows in paragraphs 15 – 26 that the invention has given rise to copying by others which has created a new industry in which millions of units has been sold. Such evidence has been indicated by the Supreme Court of the United States to be highly probative of nonobviousness of this invention. MPEP 716.01(a), 716.02(a), 716.04, and 716.06.

C. Conclusion

The Office Action presents four references, all of which do not suggest the invention. It is submitted that the large number of references that relate to but do not contain the invention in itself is a persuasive argument for nonobviousness. When combined with the objective evidence of nonobviousness presented in the Background of the Invention and the Declaration, the patentability of the claimed invention is clear.

For the above reasons, claims 57 – 66 as amended are believed to be patentable and their reconsideration and allowance are respectfully requested. No additional fee is seen to be required. If any additional fee is required, please charge it to Deposit Account No. 50-1848.

Respectfully submitted,
PATTON BOGGS LLP

By: ______________________
Carl A. Forest, Reg. No. 28,494
Telephone: 303-379-1114
Facsimile: 303-379-1155
Customer No.: 24283

VERSION WITH MARKINGS TO SHOW CHANGES MADE

In the Claims:

Claims 57 and 59 have been amended as follows:

57. (Amended) A method of producing a series of true random numbers, said method comprising:

using a hardware device to produce an analog noise signal;

converting said analog noise signal to a binary true random sequence of signals;

interfacing said binary true random sequence of signals to a general purpose personal computer; and

utilizing said interfaced binary true random sequence of signals in said computer.

59. (Amended) A true random number generator comprising:

a true random number generator circuit for generating a true random sequence of signals; and

a computer including a means for interfacing with said true random number generator circuit, said means for interfacing consisting of one or more of the following: a device driver, a TSR, a portion of the operating system of said computer, and a program stored in the bios memory of said computer.